Dynamic Programming: Problem 5

We have a directed graph G=<N, A>, where N={1,..., n} is the set of nodes and E▪NxN is the set of edges. Let M be the adjacency matrix of the graph G, that is, M[i, j]=TRUE if the edge (i, j) exists, and M[i, j]=FALSE otherwise.

You are interested in knowing from which vertices you can access any other vertex (using a path of any length), using the Warshall algorithm, so that

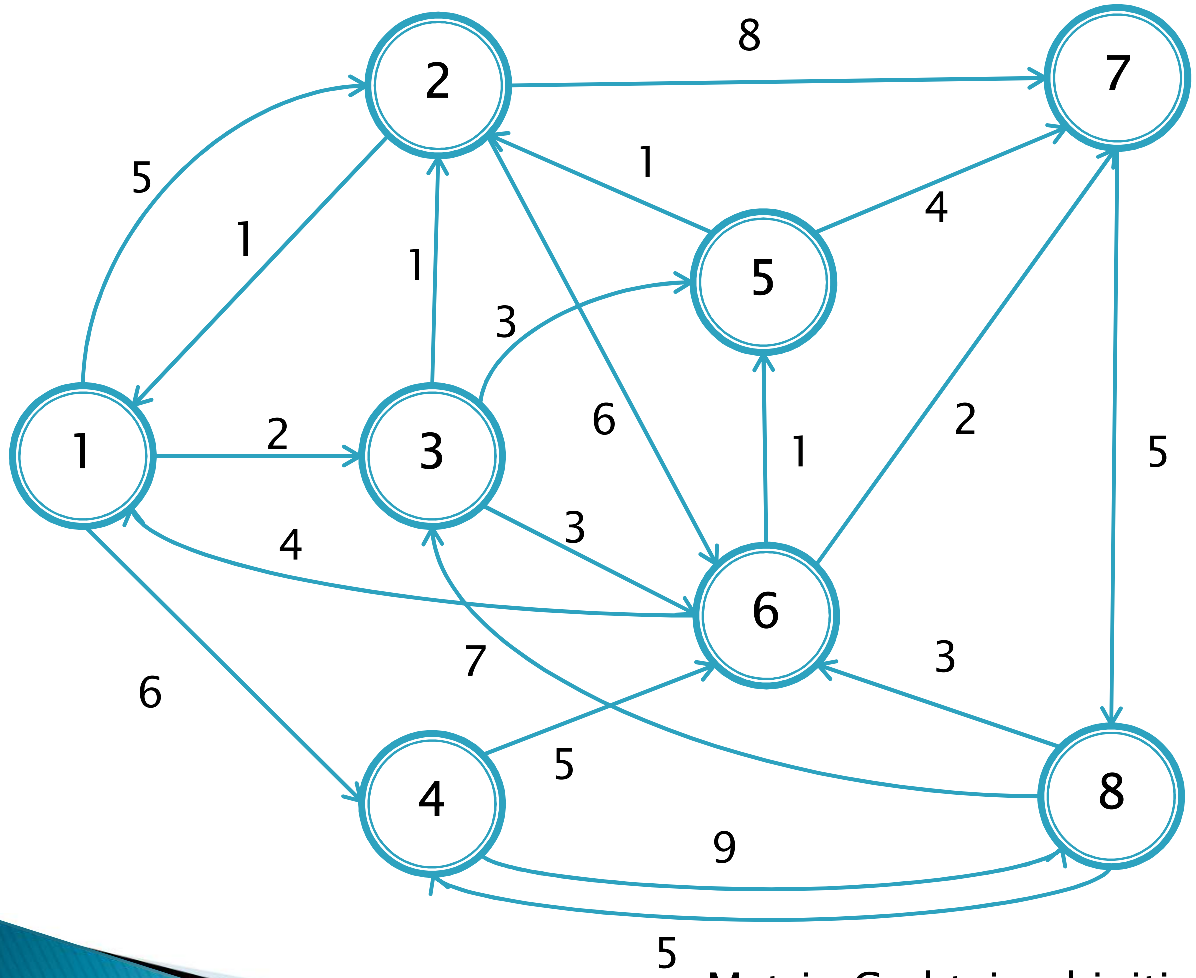
- It is necessary to obtain a matrix of paths C so that C[i, j]=TRUE if there is a path (of any length) between the nodes i and j, and C[i, j]=FALSE if there is no way to get from I to j.
- To do this, it is necessary to consider a higher number of nodes with a Bottom-Up approach. You start by trying to go directly from one node to another, then you try to find the paths that can use the vertex 1, then those that can use the vertices 1 and 2, then what you get with the vertices 1, 2 and 3, etc., until you get the paths that can use all vertices from 1 to n.

Implement a Dynamic Programming method in the above terms, which obtains the matrix C of existence of paths of a graph having as data the number of nodes n and the adjacency matrix M.

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

Matrix C obtained initially from M →

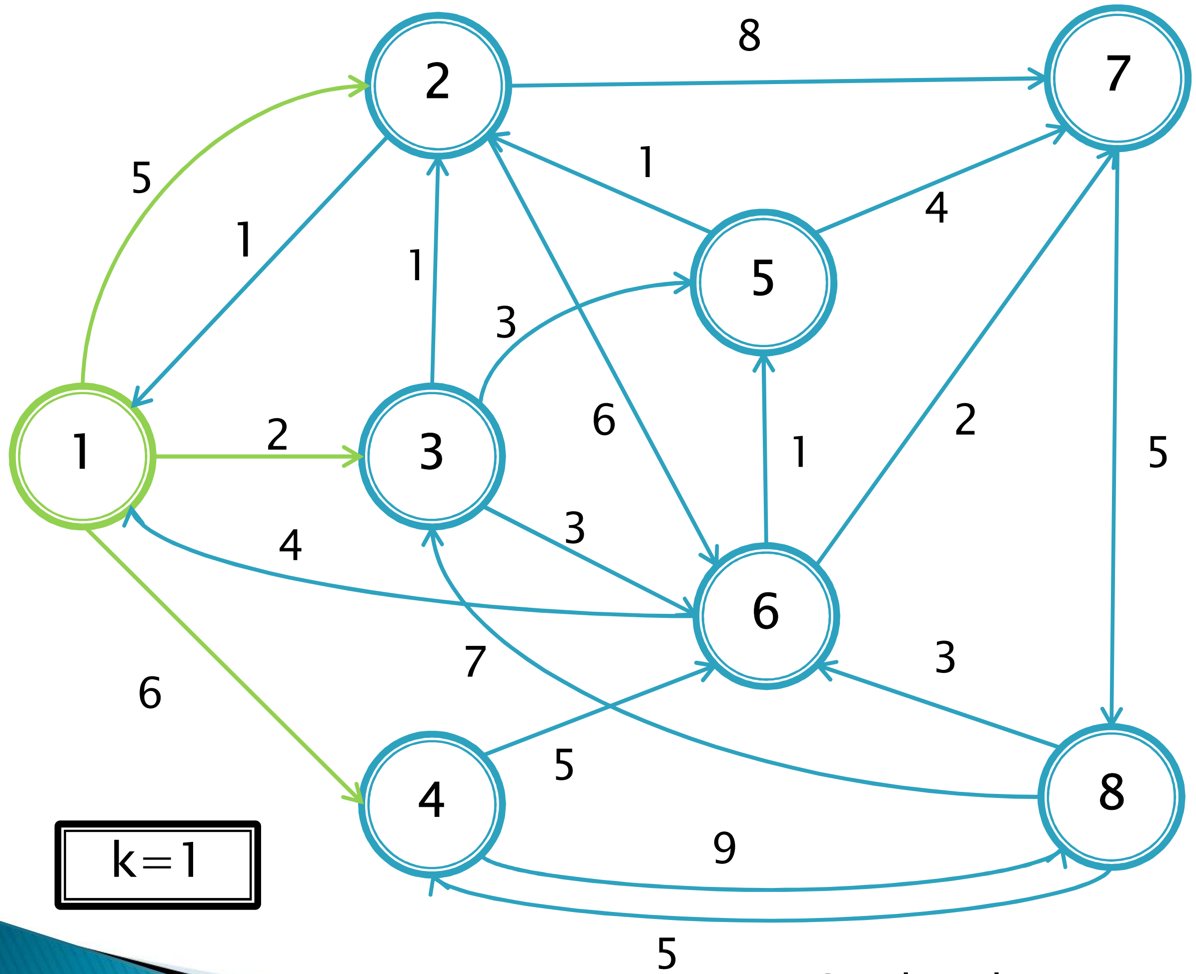
	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →

	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0



k=1

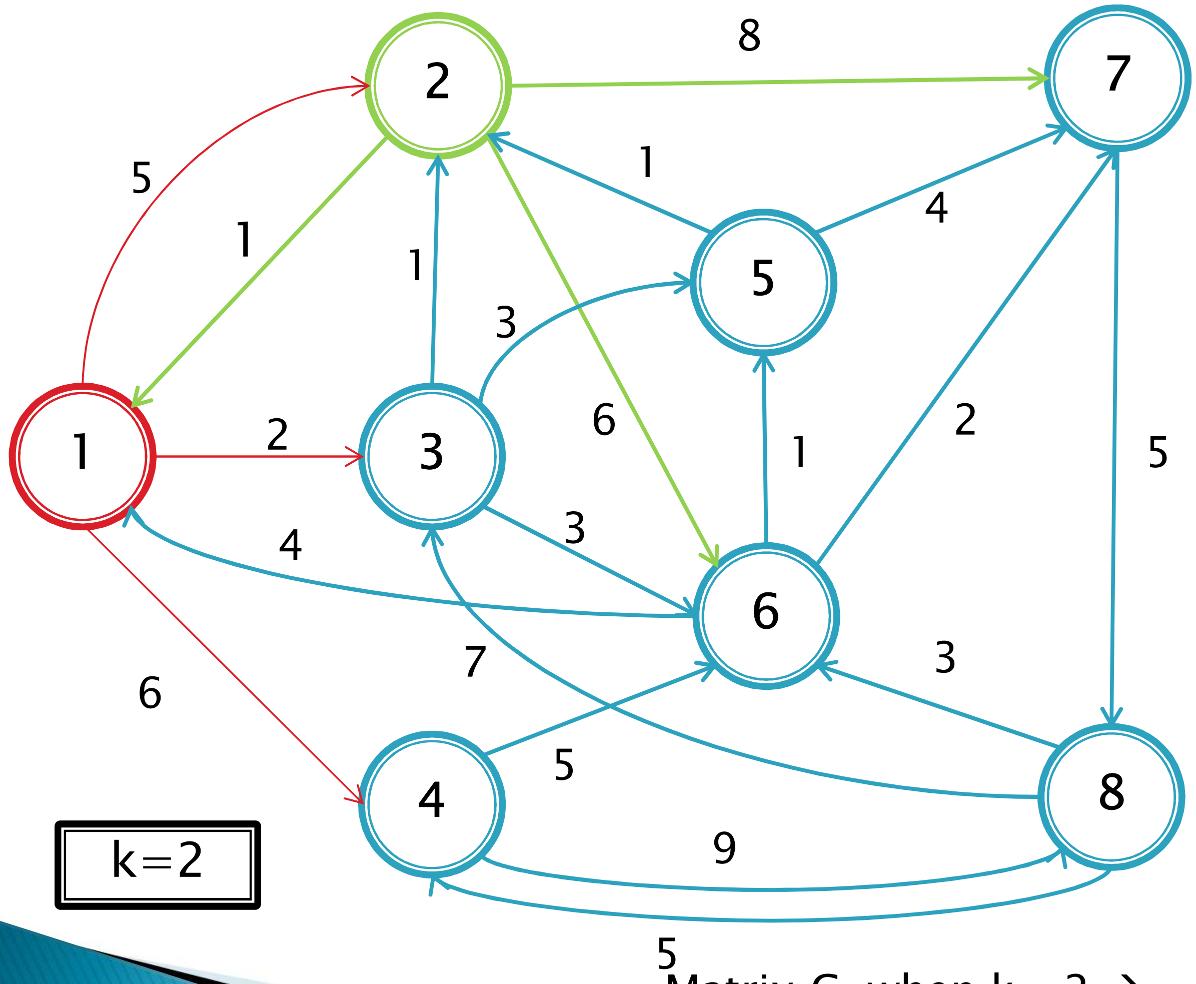
Matrix C, when k =1 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	T	T	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



k=2

	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

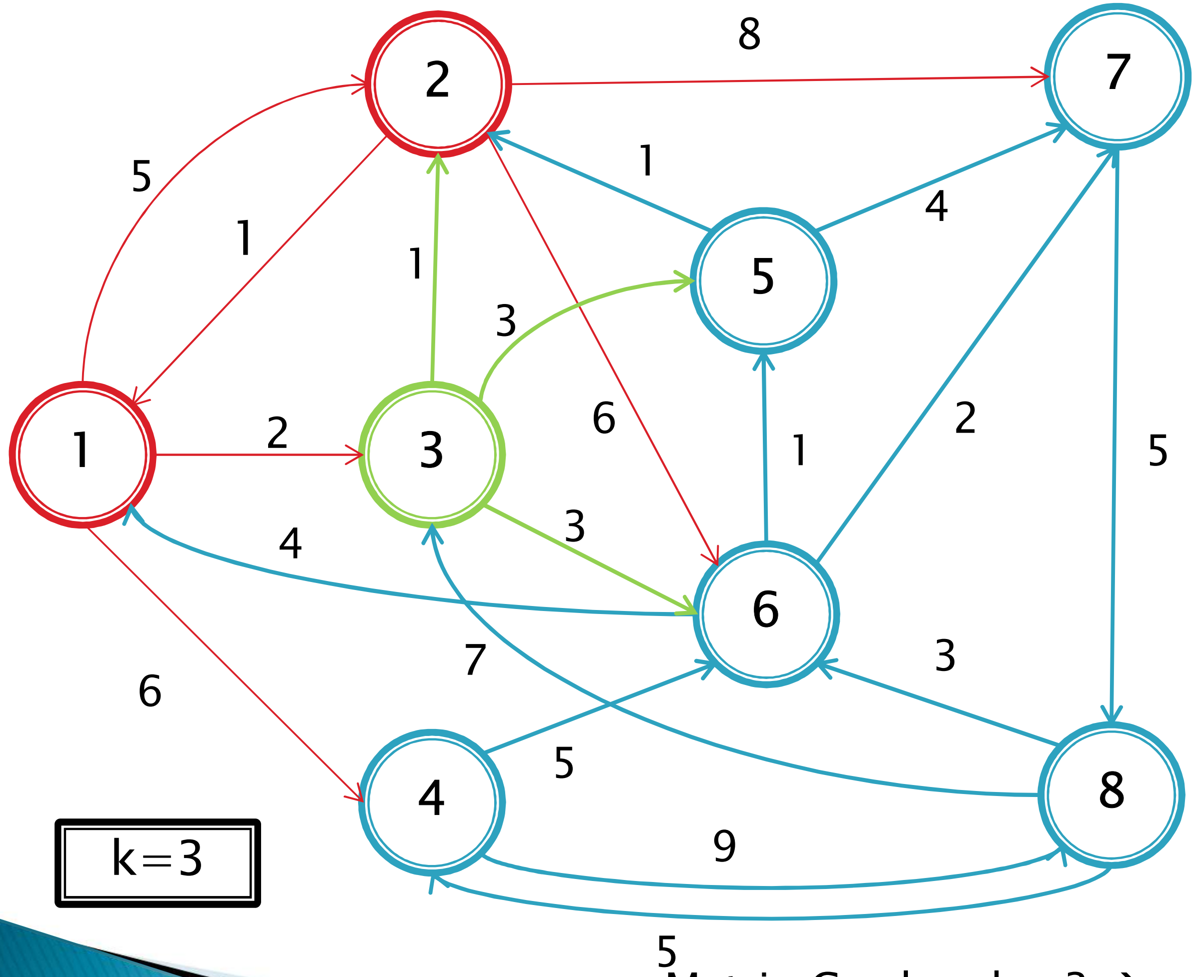
Matrix C, when k =2 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	T	T	F
2	T	T	T	T	F	T	T	F
3	T	T	T	T	T	T	T	F
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



k=3

	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

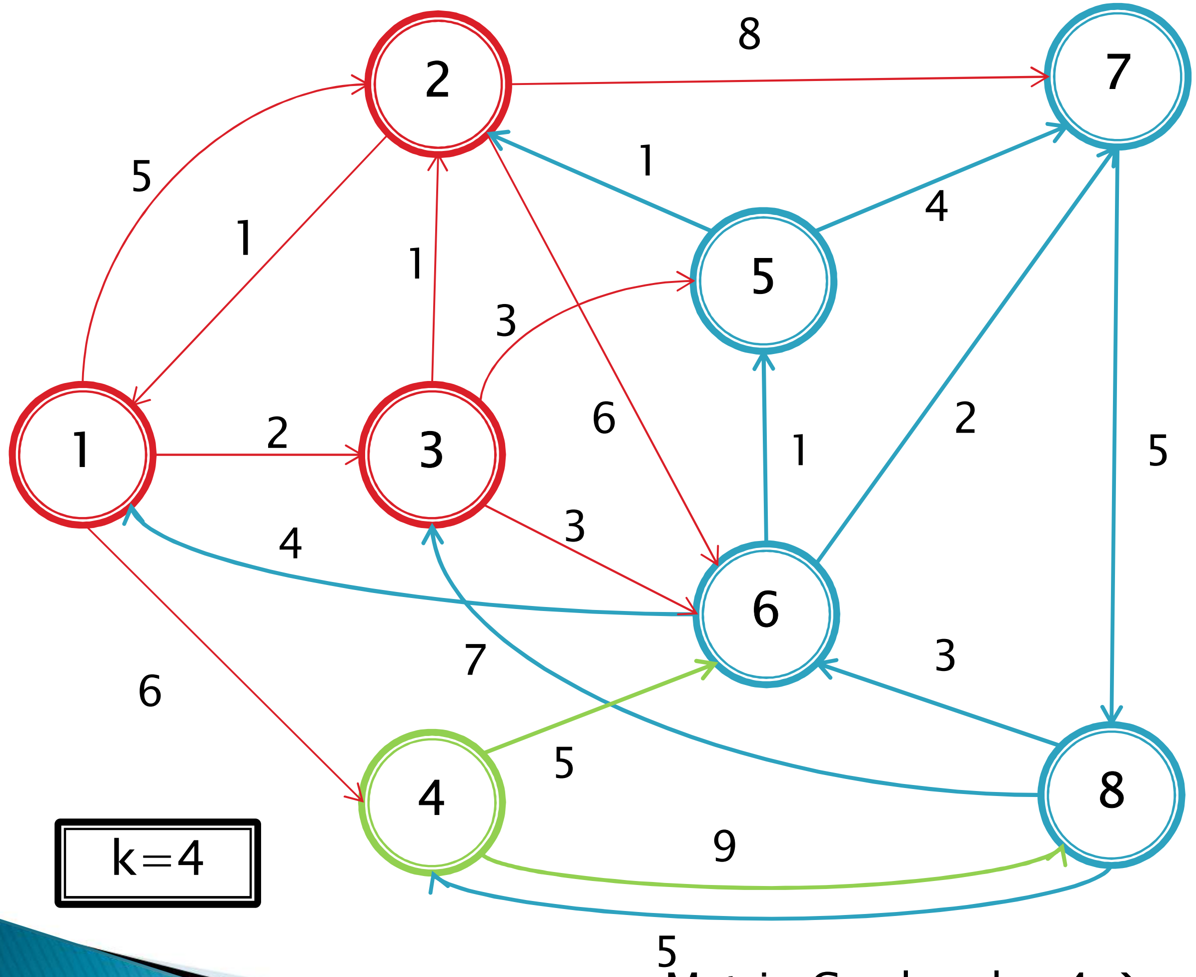
Matrix C, when k =3 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	F
2	T	T	T	T	T	T	T	F
3	T	T	T	T	T	T	T	F
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	F
6	T	T	T	T	T	T	T	F
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

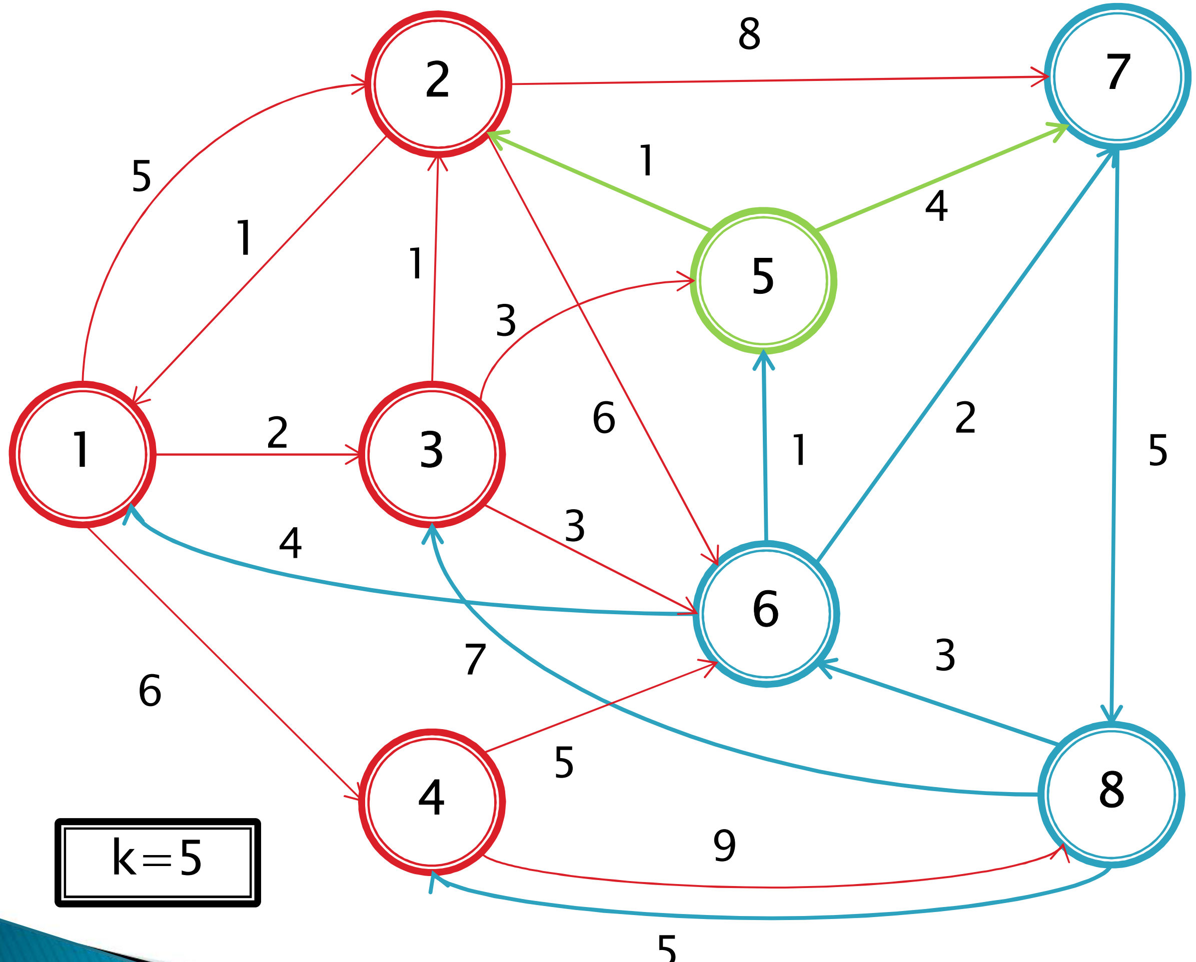
Matrix C, when k =4 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

Matrix C, when k =5 →

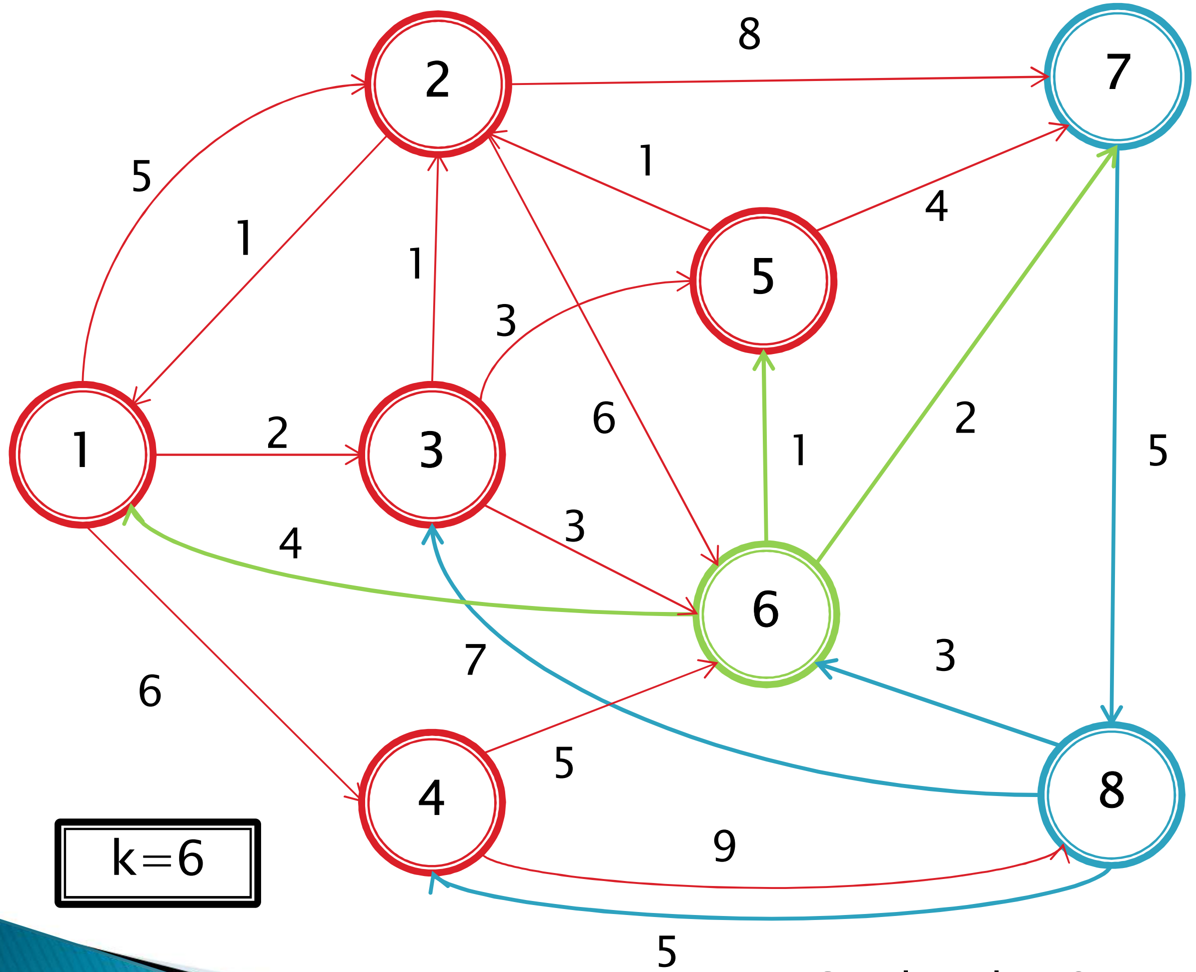
	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	F	F	F	T	F	T	F	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →

	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0



Matrix C, when k =6 →

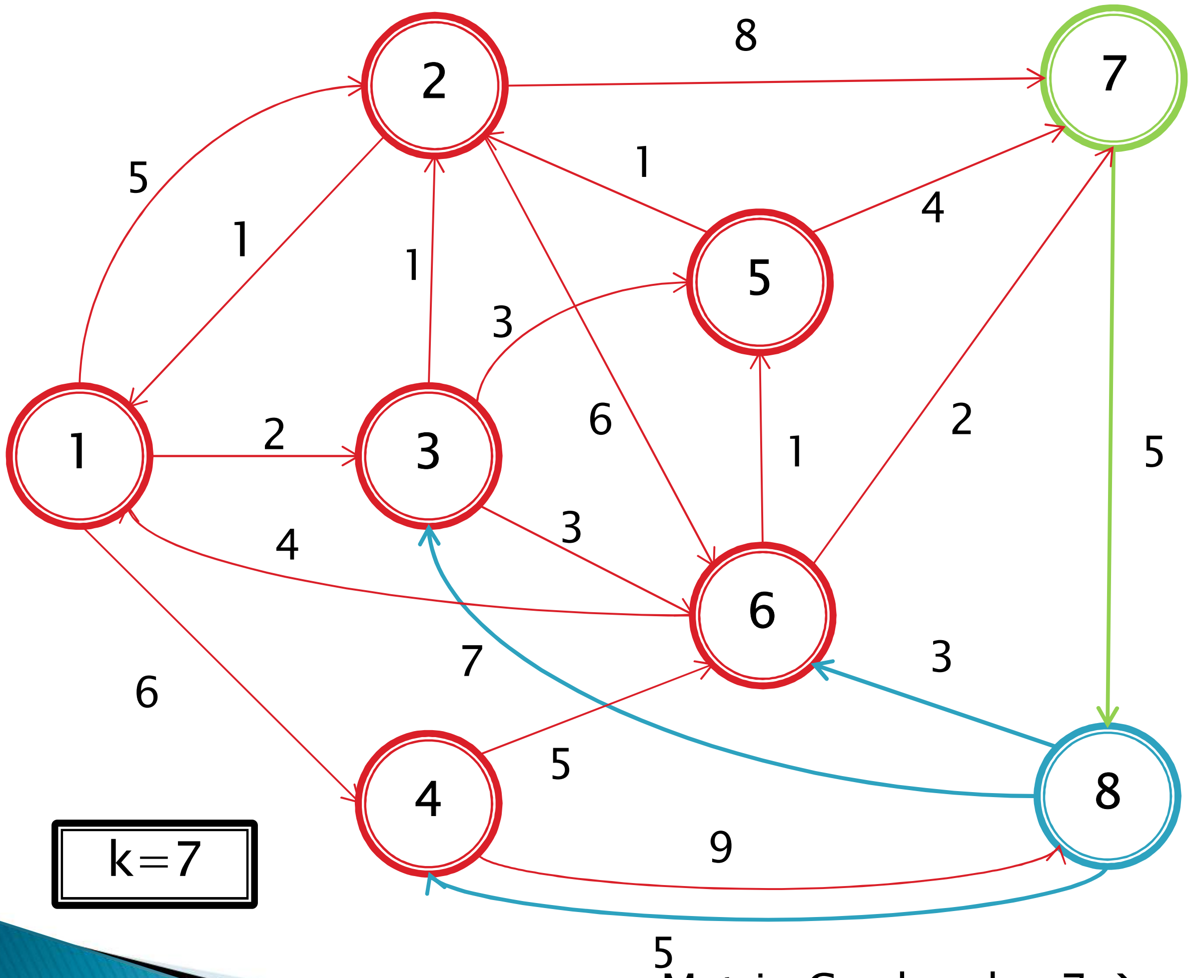
	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →

	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0



k=7

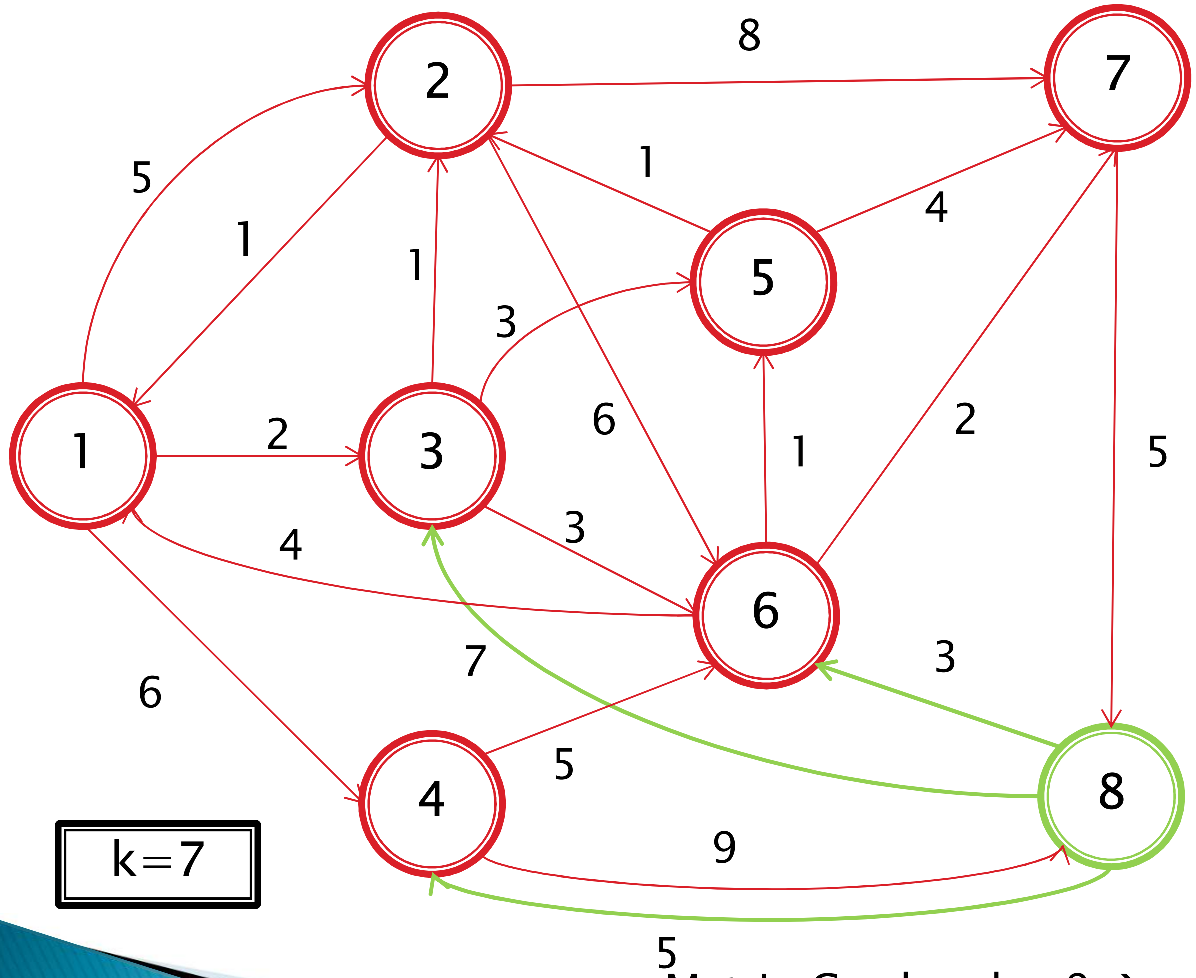
Matrix C, when k =7 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	F	F	F	F	F	F	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:

Input matrix M →



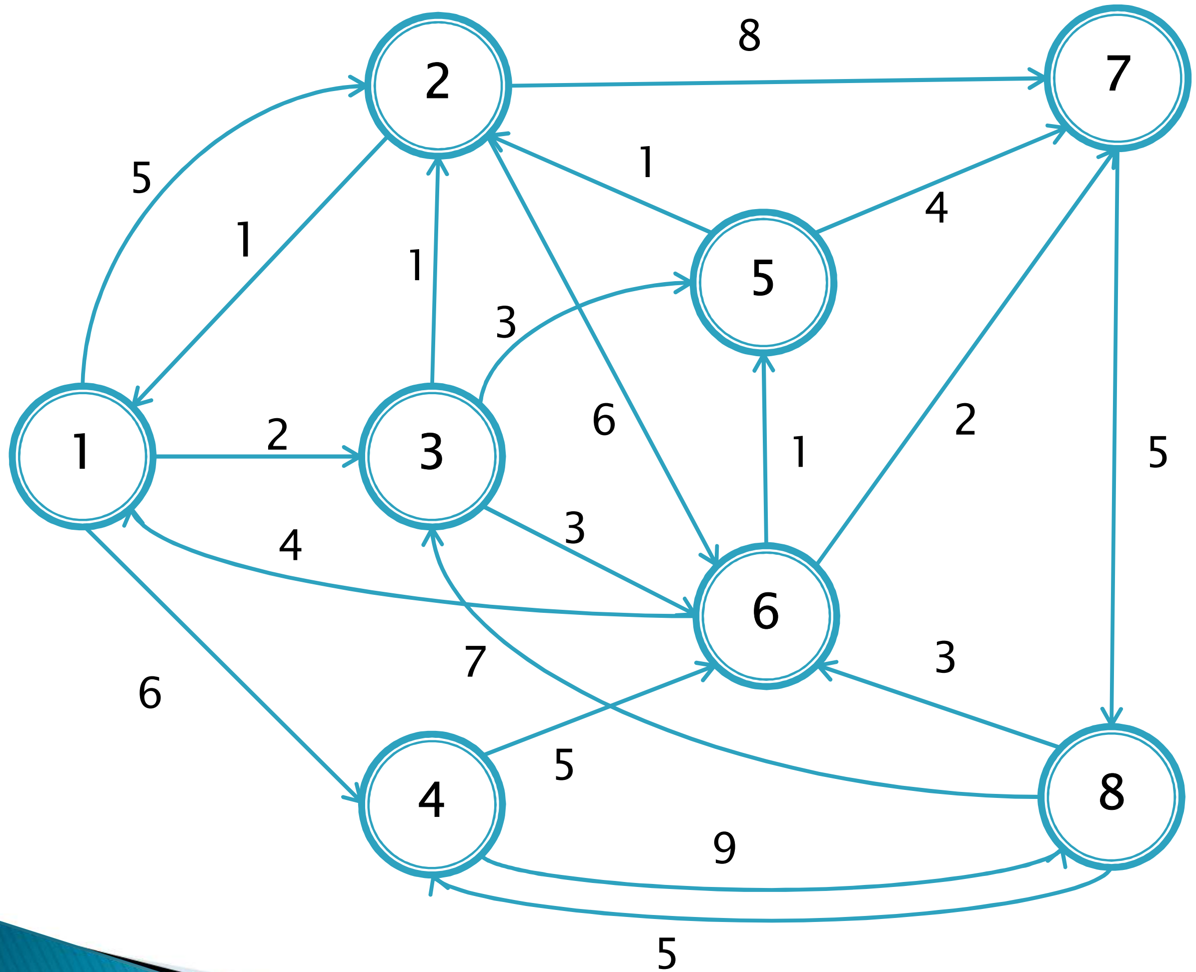
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	1	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	1	∞	∞	0	∞	4	∞
6	4	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

Matrix C, when k =8 →

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	T	T	T	T	T	T	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 1:



Starting matrix C →

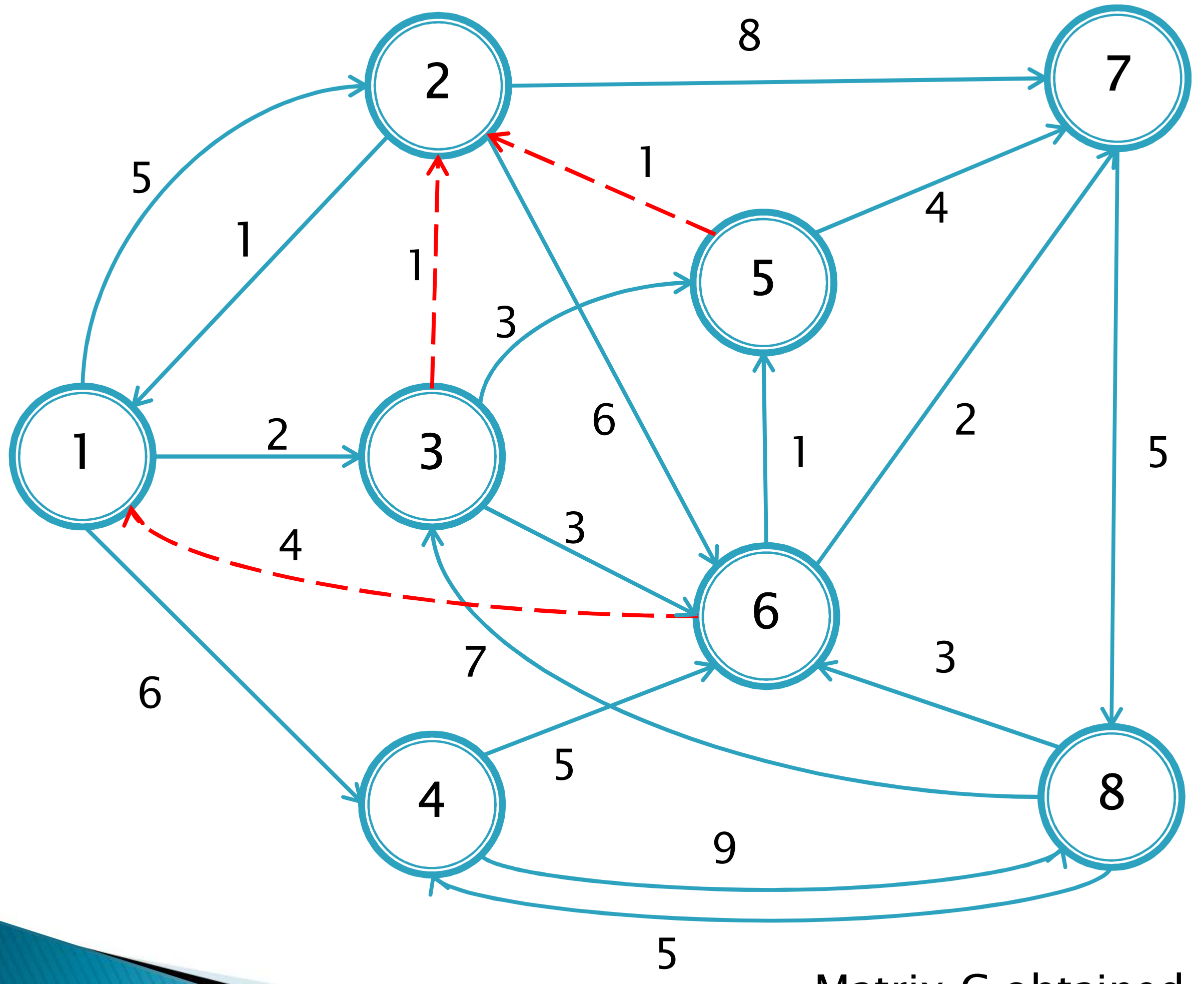
	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	T	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	T	F	F	T	F	T	F
6	T	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Matriz C obtained when→
Warshall 's algorithm is
finished

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	T	T	T	T	T	T	T	T
4	T	T	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T
6	T	T	T	T	T	T	T	T
7	T	T	T	T	T	T	T	T
8	T	T	T	T	T	T	T	T

Dynamic Programming: Problem 5

Example 2: eliminating the connnections indicated



Input matrix M →

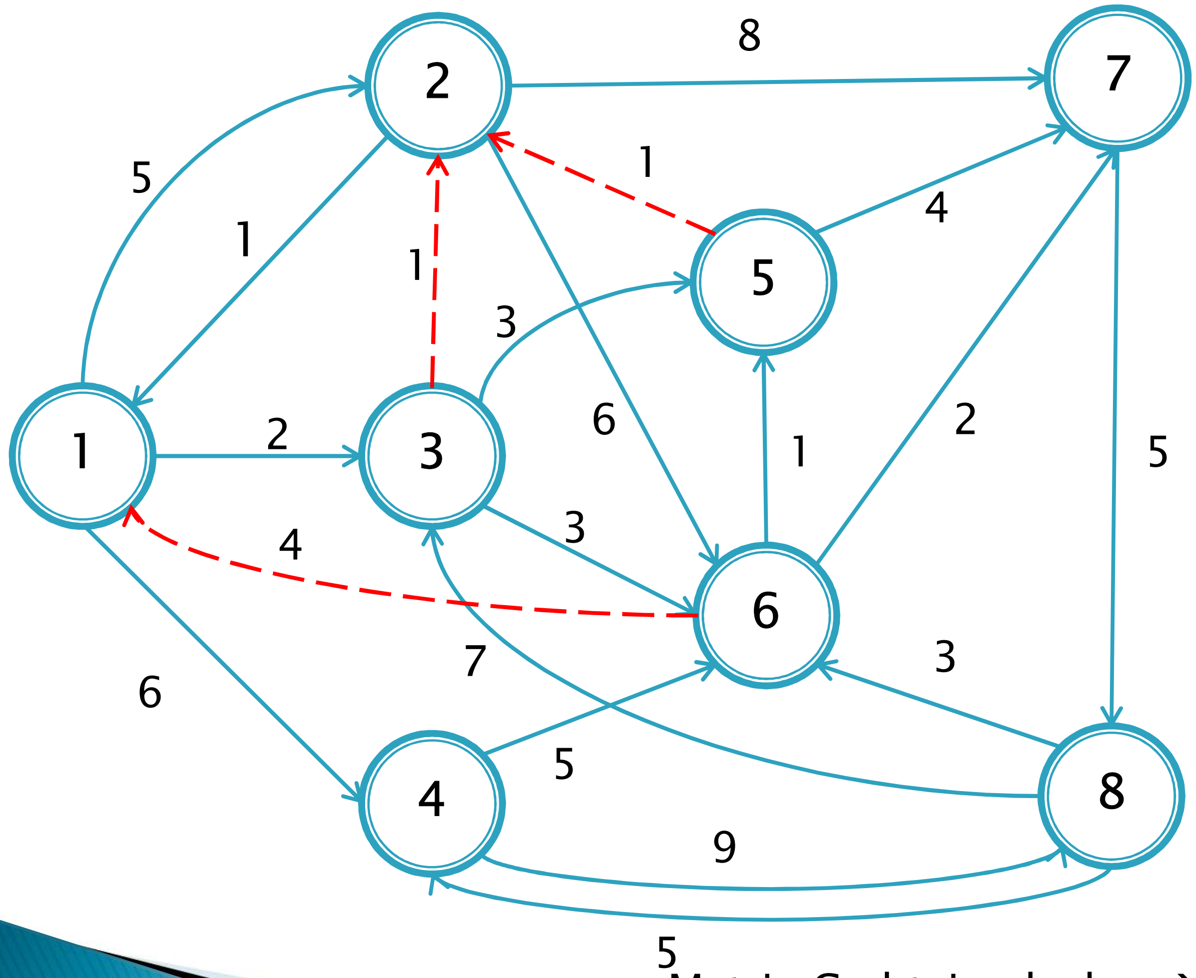
	1	2	3	4	5	6	7	8
1	0	5	2	6	∞	∞	∞	∞
2	1	0	∞	∞	∞	6	8	∞
3	∞	∞	0	∞	3	3	∞	∞
4	∞	∞	∞	0	∞	5	∞	9
5	∞	∞	∞	∞	0	∞	4	∞
6	**∞**	∞	∞	∞	1	0	2	∞
7	∞	∞	∞	∞	∞	∞	0	5
8	∞	∞	7	5	∞	3	∞	0

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	F	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	F	F	F	T	F	T	F
6	F	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Matrix C obtained → initially from M

Dynamic Programming: Problem 5

Example 2: eliminating the connnections indicated



Starting matrix C →

	1	2	3	4	5	6	7	8
1	T	T	T	T	F	F	F	F
2	T	T	F	F	F	T	T	F
3	F	F	T	F	T	T	F	F
4	F	F	F	T	F	T	F	T
5	F	F	F	F	T	F	T	F
6	F	F	F	F	T	T	T	F
7	F	F	F	F	F	F	T	T
8	F	F	T	T	F	T	F	T

Matriz C obtained when→ Warshall 's algorithm is finished

	1	2	3	4	5	6	7	8
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T
3	F	F	T	T	T	T	T	T
4	F	F	T	T	T	T	T	T
5	F	F	T	T	T	T	T	T
6	F	F	T	T	T	T	T	T
7	F	F	T	T	T	T	T	T
8	F	F	T	T	T	T	T	T

Dynamic Programming: Problem 5

- Warshall's algorithm

  1. The connections of node 1 with all the nodes are checked
  2. New connections using node 1 as intermediate are checked.
  3. New connections using node 2 as intermediate are checked.
  4. …

  N New connections using node N as intermediate are checked.

Dynamic Programming: Problem 5

- Warshall's algorithm
  - Auxiliar matrix C is used to indicate if there is a connection from node i to node j.
  - Initially:
    - C [i] [j] = FALSE si M [i] [j] = ∞
    - C [i] [j] = TRUE si M [i] [j] < ∞
  - In each k iteration ($1 \leq K \leq N$) , it is:
    - C [i] [j] = (C [i] [j]) o { (C [i] [k]) y (C [k] [j]) }

Dynamic Programming: Problem 5

```
const N = ...
type matrix= array[1... N] [1...N] of boolean
proc Warshall (I M : matrix; I/O C: matrix)
var i, j: integer
   for i=1 to N do
      for j=1 to N do
          C [ i ] [ j ] = M [ i ] [ j ]
      efor
   efor
   for k=1 to N do
      for i=1 to N do
         for j=1 to N do
            if ( C [ i ] [ k ] and C [ k ] [ j ] ) then
                C [ i ] [ j ] = true;
            eif
         efor
      efor
   efor
eproc
```